NETWORK ANOMALY DETECTION

Devashish Thakur
Stony Brook University, USA
+1-8147530089
dthakur@cs.stonybrook.edu

Kaushik Devarajaiah
Stony Brook University, USA
+1-9145366933
kdevarajaiah@cs.stonybrook.edu

ABSTRACT
In this project we try to find anomalies in a time series network traffic using two techniques. We first model the network traffic as an undirected weighted graph. Using personalized page rank we rank each IP for each day of the Network data. We try to find if the page rank of any node changed significantly across days. A significant increase or decrease in the rank may mean that the node started sending too many packets or started receiving too many packets from other servers. We model the rank of each IP in a time series and use Mean Shift model to estimate the change point. If we detect a change point we flag it as anomaly. We try the algorithm on challenge network dataset. We further try the analysis on Darpa Test Dataset and measure the prediction and recall value over the labeled dataset. Finally we run the algorithm on sipscan real-time dataset and try to find IP's with anomalous behaviors.

Another approach used is based on detecting changes in Isomorphic graph sub-components to detect presence of anomalies on a given time period by considering multiple time periods' data points.

Categories and Subject Descriptors
D.3.3 [**Programming Languages**]: Python; H2.8 **[Database Application]**: Data-Mining; C2.5 **[Local and Wide-Area Networks]**:Internet

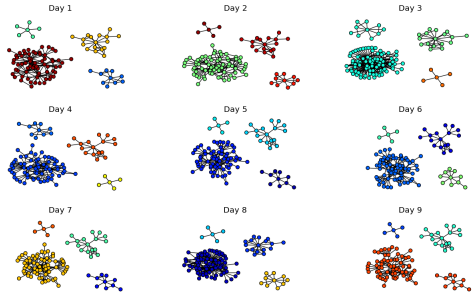
Keywords
Page Rank, Anomaly, Graph, Network, Time Series

1. INTRODUCTION

Computer networking is a critical backbone of communication systems in the contemporary world. This makes it extremely important to understand and identify irregularities in the network. The irregularities may manifest as benign outliers or harmful anomalies and it is important to distinguish the difference. Network anomaly detection like most data mining approaches is a fuzzy problem and may not have a fixed solution, as the types of intrusions change and evolve constantly. Thus, there is a need to develop solutions that are proactive in their approach in detecting unforeseen network intrusions. Network data is generally guarded so that it does not reveal unnecessary information to the outside world that could compromise the network. This makes the problem of anomaly validation fuzzier since there is hardly any ground truth to verify against.

Networks are naturally modeled as graphs, with nodes representing the entities communicating and edges represent channels of communication, which can either be duplex or full-duplex and they are represented by directed and undirected edges respectively (or optionally as a multi-graph if it is important to have retain the directionality of communication). This model is network is also native to computing systems as various computing algorithms are modeled as graphs. With this representation of networks, the parameters that we can use to gauge a network are nodes, degree (out/in or both), edges in he network graph, connected, bi-connected subcomponents, articulation points and others. Apart from these basic graph parameters, it is possible to include other information specific to networks such as, connection/packet information, protocols used, frequency of communication, geo-location of a node etc.

In our work, we focus on identification of anomalies in graph data by leveraging basic graph parameters above, so that it is generic enough to be applied to other graph data. However, there are advantages in considering specific network related parameters to fine-tune results, which is beyond the focus of our approach. We use techniques called as page-rank and properties such as graph isomorphism to target the problem of anomaly detection. We discuss our approaches and demonstrate the results to show the validity of the same.



Day-wise component of [1] dataset

The figure above shows the insights that can be gained by a graph representation of network data. It clearly shows a noticeable pattern across each day. The data is taken from [1].

We used R, Python libraries such as Network and Matplotlib and SciPy and for various analyses on the graph data. Most of the basic algorithms involved are linear in the size of data involved except for graph isomorphism checks.

2. RELATED WORK

A lot of work has already been done in this field. People have used numerous methods to find anomalies in network data. Many researchers have tried to pinpoint the exact IP, which is anomalous; some have tried to analyze the behavior of anomalous IPs and the feasibility of network anomaly detection in real-time network. [4] Gives an introduction to various network anomaly

attacks and scalability issues in detecting those attacks. The authors in [5] apply K-means algorithm to find anomalies in network traffic data. The approach is simple and scalable, however the author does not suggest any method to find the optimum K. Applying clustering each time for different value of K has scalability issues. [6] Uses PCA to find K-Eigen vectors and tries to cluster data along those Eigen values. [7] Tries to model data in the form of a graph and tries to find entropy changes in the attributes (source and destination IP and Port) graph. The authors of [8] apply hierarchal clustering on traffic data and then apply SVM on individual clusters for classification of data. As a result SVM is not applied on the entire dataset, which hides the scalability issues of SVM. [9] Compares the neural network and decision tree algorithms to find network anomalies. While neural networks can find known attacks easily, decision tree was able to find unknown attacks easily. The authors do not provide any method to improve the accuracy of neural networks so that it can efficiently find unknown attacks as well. The authors in [10], [11] and [12] use graph based anomaly detection. In [10] the author tries to find the most isomorphic substructure in the graph. [11] Tried to find anomalies in the graph using neighborhood formation. It tries to find nodes linked to other nodes, which are not each other's neighbor. The authors in [12] try to find black hole and volcanoes in a network and they term those as anomalies in the graph. The authors in [13] apply co-clustering to detect anomalies using a combination of soft and hard clustering methods called Sparse Matrix Regression and Information Theoretic co-clustering respectively.

In [14] authors use Single-linkage clustering considering all connection parameters to pinpoint anomalies. In [15] supervised approaches such as classification, link analysis and sequence analysis are utilized. It also uses some domain knowledge to find anomalies whereas [16] employs a technique called Penalized Basis Pursuit involving PCA and EMA (Exponential Moving Average) for anomaly detection. In [17] authors uses Random forest algorithm for the network anomaly detection, with the assumption that majority of data is attack free. [18] Uses two sub-techniques called as anomalous substructure detection and anomalous sub-graph detection. The key idea portrayed by the authors here is to use compressibility as a factor for detecting anomalies.

3. PROPOSED METHOD

3.1 Page Rank and Mean Shift algorithm

We try to represent the network data in the form of a graph. Once we have done that we try to find page rank of each IP across days. We use page-rank algorithm for this with teleportation factor of 0.9. A high page rank is related to the number of nodes pointing to it.

Once we have modeled this we use Mean shift model proposed in [21][23] to find a shift in the time series. In this method we first normalize the rank value for each IP. The method then attempts to detect shift in the mean of the time series using a variant of mean shift algorithm for time shift analysis.

To find the change in time series the algorithm first creates a CUSUM chart by calculating the cumulative of all the points. Let this be S_{diff} it then does bootstrap analysis by randomly reordering the data. The aim is to mimic the behavior of CUSUM if no change is occurred.

A number of bootstrap is constructed and their CUSUM is calculated. Let it be $S\acute{b}_{diff}$. Then the confidence level is calculated using the formulae $100 * \frac{X}{N}$ where X is the number of bootstraps for which $S\acute{b}_{diff} < S_{diff}$ and N is the number of bootstraps. A confidence score of 90-95% confidence is required before it can be stated that a change has occurred.

Once a change is detected the algorithm then tries to find when the change has occurred. For this it uses Mean Square error (MSE).

MSE (m) = $\sum_{i=1}^{m}(X_i - \bar{X}_1)^2 + \sum_{i=m+1}^{N}(X_i - \bar{X}_2)^2$

Where $\bar{X}_1 = \frac{\sum_{i=1}^{m} X_i}{m}$ and $\bar{X}_2 = \frac{\sum_{i=m+1}^{N} X_i}{N-m}$

The value of m that minimized MSE (m) is the best indicator of the last point before the change. The point m+1 is the first point after the change. The threshold of MSE (m) is provided in the algorithm. A high threshold may find less anomalies but the accuracy or precision is higher. A low threshold may find more anomalies with a lower precision

3.2 Sub-component Isomorphism Method

We borrowed the approach of anomaly detection using least representative substructure from [24]. For this, we used isomorphism-based method to detect presence of anomaly. However, the results were not satisfying since many points were chosen as anomaly for the dataset we chose. So, we extended that method by the following steps 1) Identify connected components in the graph. 2) Among 1, find set of components, which are isomorphic. This provides insights into their communication neighborhood structure. 3) Find metrics on 2 such as, set size, size of the isomorphic sub-connected components. 4) Fit a line for the plot of above metrics for each day and use the slopes across various days to find anomalous activity over different days. We use an algorithm based on change-point from [23] to pick outliers in the array of slopes.

Step 1 and 2 were based on [24]. The intuition for 3 was based on the assumption that a discerning pattern is noticeable over days and anomalous activities over days could be captured by comparison of day level data. We plot the day-level metrics - Isomorphic set size and size of each isomorphic connected component that form the set.

We tried PCA on these 2 attributes to reduce dimensions of the day-wise sets, in order to compare them across days, but it was not fruitful as the projection into a single dimension after PCA for the sets resulted in values that did not make the anomalies stand out. We discovered that this was due to the loss of information due to PCA. But we resorted to a simpler approach of fitting a line through these data-points and using the slope to compare days. The intuition was that anomalous IPs changed graph connectivity and in-turn would reflect in the reorientation of day-wise metrics for the anomalous day.

4. Experimental Results

4.1 DARPA Dataset [19]

We tried the proposed method first on labeled DARPA Intrusion detection dataset. Table below describes the dataset

Table 1 – Description of DARPA dataset

Feature	Value
# Days of data present	13
# Unique IP Addresses	3303
# Lines of data	13,75,739
# Labeled anomalies	253

We ran out page rank algorithm on the dataset for each IP for each day. We then used Mean Shift Algorithm [21] to find the IP that shows significant change. We use a threshold of 0.81 as the Mean shift error.

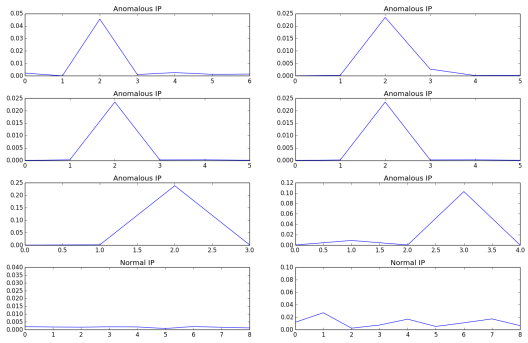


Figure 3 – Anomalous and non-anomalous IPs of Test Data

Figure 3 shows the graph of page rank vs. days for 6 IPs. The first four are anomalies whereas the last two are normal data. As we can see in the graph plotted above, the anomalies show a spike in their time series graph. We compared our results with a random classifier and most frequent classifier. Using [20] we calculated the precision and recall of our results.

Algorithm	Precision	Recall
Randomized Classifier	50%	50%
Most Frequent Classifier	53.90%	0%
Proposed Method	69.58%	84.88%

The proposed method labeled 419 IPs as anomalies and 2889 as non-anomalies. Out of 419, 219 anomalies were correctly labeled as anomalies.

4.2 Synthetic Dataset

We tried the subcomponent algorithm method on synthetic data set generated artificially through code.

We created a graph with 1000 nodes. On some random day we increase an anomalous node. We then plot the graph of subcomponents size vs Number of component for each day and use Linear Regression [20] to find the best-fit curve for the graph. We measure the slope of the graph. We do the same for all the days. We observe that the slope of the graph on the day of anomaly was considerably higher than that of the other days. The exact day when the slope changes can be found by Mean Shift algorithm [21][1].

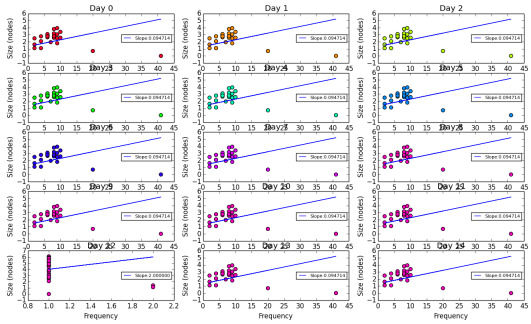


Figure 10 – Maximum subcomponent plotted for 12 days

The above figure 10 shows the graph for 15 days. We inserted the anomaly on 13th day and the slope of the graph was observed to 2.01, whereas every other day had the slope of 0.9-0.8. So using Mean Shift algorithm [21] we can pinpoint the particular day, which has anomalous data.

4.3 SIPSCAN Dataset [2]

We next tried the proposed page rank algorithm on SIPSCAN dataset. The below table describes the dataset.

Table 1 – Description of SIPSCAN dataset

Feature	Value
# Days of data present	10
# Unique Source IP Addresses	2,954,108
# Unique Destination IP Addresses	14,534,793
# Unique couples (source IP - destination IP)	20,241,109
# Lines of data	2,02,55,722
Max # of destination a source targets	17613
Average # of destinations a source targets	6.85
Max # of sources targeting a destination	14
# Of Source IPs present on at-least 3 days	8803

Before we started working on the dataset we tried to find the degree distribution of the sipscan [2] data set for all the days.

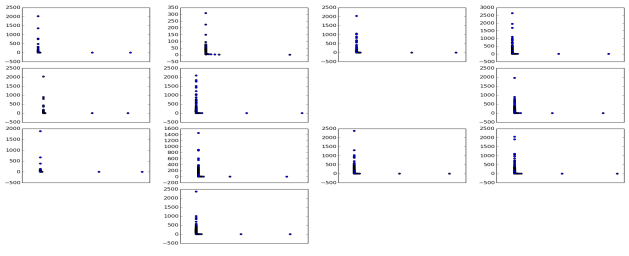


Figure 1 – Frequency Vs. Degree

On analysis of Figure 1 we see that the data is very sparse and the graph obeys power law. The number of nodes with high degree is very less and the number of nodes with very low degree is very

high. The number of nodes with high degree is very less on Day 2, 5, 9 and 13, where as Day 3,4 and12 has a high percentage of nodes in the higher degree range.

Next we tried to find the redundancy coefficient of the data. Redundancy coefficient of a node V shows the fraction of neighbors of V that is linked to another node other than V. If the bipartite graph were projected [3] then these neighbors would remain connected even if V were removed. So if the redundant coefficient were 1 then the projection would remain the same even without V. If the value is 0, then it means that neighbors of V are not connected with anyone other than V.

The formulae for calculating redundancy coefficient of a node V in a graph is

$$C(V) = \frac{\left|\{\{u,w\}\ is\ a\ subset\ of\ N(V), \exists\ v' \neq v, (v',u) \in E\ and\ (v',w) \in E\}\right|}{(|N(V)|)(|N(V)-1|)/2}$$

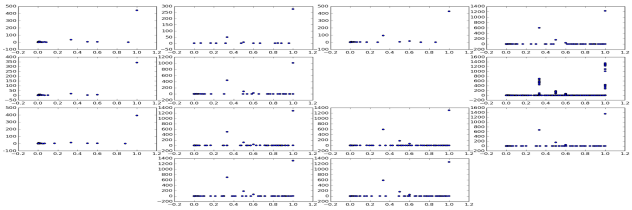


Figure 2 – Frequency Vs. Redundancy Coefficient

In Figure 2, we see a cluster of nodes (~50) on Day 2 with coefficient ~0.4 and this number increases in Day 4 and Day 6. Day 8 sees a drastic increase in the number of nodes with coefficient around 1. This means that the number of neighbor of the nodes increases in Day 8 which means that the graph connectivity increased on Day8. Day 9 to 15 is almost similar with majority of cluster lying between 0.3 and 0.4.

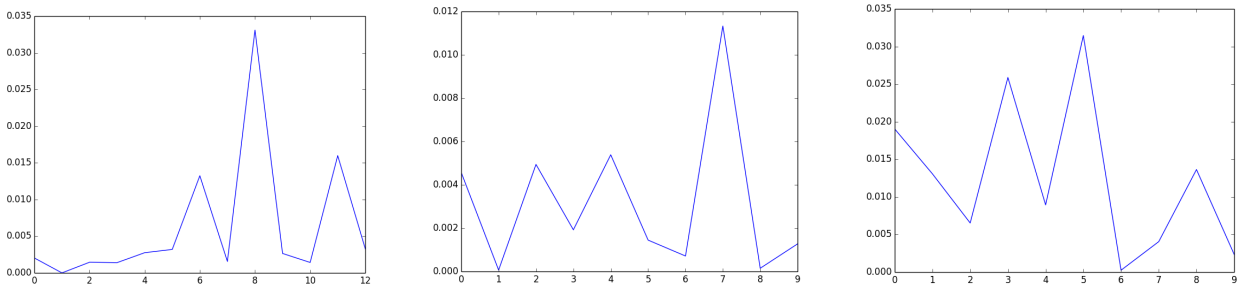


Figure 5 – Rank VS Days for Anomalous IP – 0.7.0.3, 0.1.65.85 and 0.0.80.98 for SIPSCAN Data

After the initial analysis we then tried our page rank algorithm on each day. We use Network [22] page rank algorithm to calculate the page rank of each node.

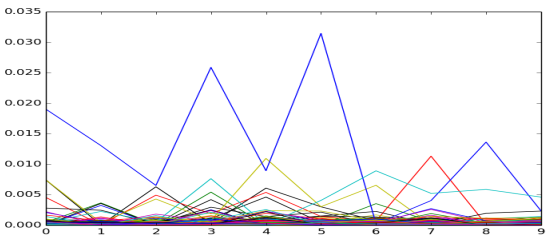


Figure 4 – Rank Vs. Number of Days

Figure 4 shows rank of 8803 IPs of SIPSCAN data for 10 days. We see a number of IP with considerable shift in their page rank. We use Mean Shift [21] algorithm to find the change points in the graph.

We find a total of 469 anomalies in SIPSCAN. Figure 5 shows some of the anomalies found. We can see a considerable drift in the time series in all the graphs. The rank of 0.7.0.3 goes from 0.03 on Day 5 to 0.00001 on Day 6 and then goes to 0.01 on Day 8.

Figure 6 shows the time series graph of a non-anomalous IP. We can see that the rank of the graph remains between 0.0 and 0.0004. The change isn't considerable and so is not marked as an anomaly.

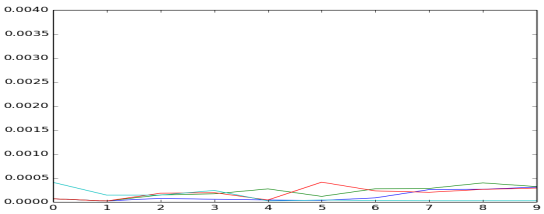
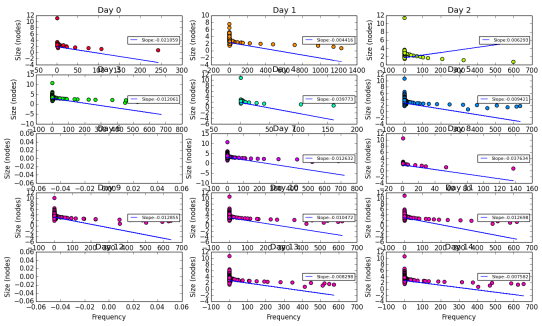


Figure 6 – Rank Vs. Days for Non-anomalous IP

We also ran Method 2 to find anomaly day on sipscan dataset on a subset of 100000 points per day, due to the hardware limitation. The result follows.



Day 3 pointed out as anomaly when run on Sipscan data on a set of 100k data points per day.

5. CONCLUSIONS AND FUTURE WORK

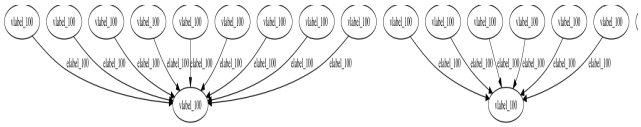
In this paper we proposed two algorithms for finding anomalies in the graph. Using page rank and mean shift method we could find IPs, which are responsible for anomalous activities. We tested the above algorithm on DARPA dataset and achieve ~70% accuracy. We also tested the algorithm on a much bigger dataset and could find IPs having anomalous behavior.

We also proposed a second algorithm, which is based on maximum subcomponent isomorphism method to find presence of anomalies on a given day. Using this method we can find the day that had anomalous behavior. We currently cannot pinpoint the IPs, which had anomalous behavior using this method. So, IP level anomaly detection is one possible extension that can be done to this approach. We tried this algorithm on synthetic dataset and we could always find the day when the anomalous dataset was introduced. We also ran this on a subset of Sipscan data and obtained a result pinpointing an anomaly day based on the difference of the isomorphic components.

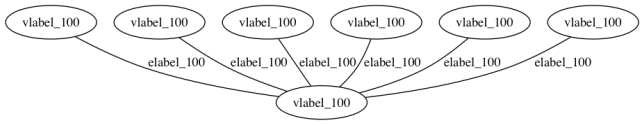
6. EXPLORATION

Below are the methods we applied but couldn't get results from it.

The data was also analyzed using SUBDUE [24]. A section of the graph is shown below.



Application of Subdue was done on a sample of 1000 edges and the most dominant repeating substructure is below.



The above substructure minimizes the Description Length needed to encode the entire graph. SUBDUE does not support finding the worst substructure, which is needed to identify anomalies. Our approach extends the method of [24] which basically uses isomorphic components for anomaly detection.

7. REFERENCES

[1] http://www.cs.stonybrook.edu/~leman/courses/14CSE590/data/challenge_network.zip

[2]

[3] The CAIDA UCSD Network Telescope on the Sipscan Dataset, http://www.caida.org/data/passive/sipscan_dataset.xml

[4] Basic notions for the analysis of large two-node networks - By Latapy, Magnien and Vecchio

[5] On the difficulty of Scalable Detecting Networks - By Levchenko, Paturi and Varghese

[6] Traffic Anomaly Detection using K-means Clustering - By Gerhard and Georg Carle

[7] In Network PCA and Anomaly Detection - By Ling Huang, Nguyen, Minos, Jordan, Joseph and Nina Taft

[8] Using Graph to Detect Network Anomaly - Zhou, Guangmin, Weisong

[9] A Novel intrusion detection system based on Hierarchal clustering and support vector machines - By Shi-Jinn, Ming-Yang, Yuan-Hsin, Tzong-Wann, Rong-Jian, Jui-Lin

[10] Neural Network vs. Decision Trees for Intrusion Detection - By Yacine Bouzida and Frederic Cuppens

[11] Discovering Structural Anomalies in Graph-Based Data - By William Eberle and Lawrence Holder

[12] Neighborhood Formation and Anomaly detection in Bipartite Graph - By Jimeng, Huiming, Deepayan and Christos

[13] Detecting Blackholes and Volcanoes in Directed Networks - By Zhongmou Li, Hui Xiong and Yanchi Liu

[14] Evangelos E. Papalexakis†, Alex Beutel†, Peter Steenkiste – "Network Anomaly Detection using Co-clustering".

[15] Leonid Portnoy – "Intrusion detection with unlabeled data using clustering"

[16] Wenke Lee, Salvatore J. Stolfo, Kui W. Mok – "A Data Mining Framework for Building Intrusion Detection Models".

[17] Brian Eriksson, Paul Barford, Rhys Bowden, Nick Duffield, Joel Sommers, Matthew Roughan – "Basis Detect: A Model-based Network Event Detection Framework".

[18] Jjong Zhang and Mohammad Zulkernine - "Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection"

[19] Celeb C Noble, Diane J Cook - "Graph-Based Anomaly Detection" [3] Charu C Aggarwal, Philip S Yu -"Outlier Detection for High Dimensional Data"

[20] DARPA Intrusion Detection Evaluation Data Set http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/1998data.html

[21] SciPy Learning Kit - http://scikit-learn.org/stable/auto_examples/plot_precision_recall.html

[22] W. A. TAYLOR. Change-Point Analysis: A PowerfulNew Tool For Detecting Changes

[23] Exploring Network Structure, Dynamics, and Function using NetworkX – By Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart

[24] Statistically Significant Detection of Linguistic Change - By Vivek Kulkarni and Steven Skiena

[25] Graph-Based Anomaly Detection - Caleb C. Noble , Diane J. Cook

[26] SUBDUE Graph based knowledge discovery - http://ailab.wsu.edu/subdue.